Dashboards

Previous: subtotals

Dashboards in Crunch

You can create a customized dashboard using the `crunchy` library (which is based on R's `shiny` library) and then embed that so the app displays in your Crunch dataset's Dashboard view. In particular, Crunch's Dashboards are a good place to incorporate non-survey data that you may want to bring in as an additional key indicator, for example sales numbers from a separate Finance database. What you can do with a shiny app is extremely broad and there are inspiring examples around the web.

A Minimum Working Example

How to create a shiny dashboard is its own deep topic with many resources at your disposal and so we will not cover it here. For this vignette, we will use the server.R and ui.R files below, which serve as a minimum working example.

```
# ui.R
library(crunchy)
library(shinydashboard)
shinyUI(
  dashboardPage(
    dashboardHeader(disable=TRUE),
    dashboardSidebar(
      selectInput(inputId = "pid", label="Political Party", choices=list("Democrats"=1, "Independents"=
    ),
    dashboardBody(
      crunchy:::loadCrunchAssets(),
      crunchy:::crunchAuthPlaceholder(),
      fluidRow(
        box(plotOutput("plot_age", height="250"), width = 6)
      )
    )
  )
)
```

```
# server.R
library(crunchy)
options(httpcache.log="")
shinyServer(function(input, output, session) {
    dataset <- shinyDataset("Economist/YouGov Weekly Survey")
    in_pid <- reactive(if (length(input$pid) == 0) c(1:3) else as.numeric(input$pid))
    ds <- reactive(dataset()[dataset()$pid3 %in% in_pid()])
    output$plot_age <- renderPlot({
      par(las=1); par(mar=c(2,2,2,0))
      hist(as.vector(ds()$age), main='Age')
    })
})
```

Running a crunchy app locally

We want to run our crunchy app locally to make sure it looks like what we want. First, in your R session, set the working directory to wherever you have saved your ui.R and server.R scripts using the `setwd()` function. Then, we will run the line below which actually builds the app.

```
# Note, you may need to install these libraries before running the app for the first time
# install.packages('crunchy')
# install.packages('shinydashboard')
runApp(, port=4747) # specify any 4 digit number
```

One last step to view the app. In a web browser, open a new tab and go to `http://local.crunch.io:4747` to see the output. Note that the 4 digit number will be whatever you specified in the port option above (4747 here).

Embedding a crunchy app in Crunch

Once you are happy with how your crunchy app looks there a few remaining steps to have it appear in Crunch itself: loading your scripts onto the server, and displaying your app in the Dashboard view.

Loading your crunchy app scripts onto the server

Log into the Crunch webapp, then open any dataset. Once the dataset is open, click on the name of the dataset in the upper left corner of the screen, which generates a menu, and click on Notebooks, which will open a new browser tab.

Next, follow the directions here to load the scripts onto the Crunch system.

Displaying your app in the Dashboard view

First, copy the url you created in the step above, e.g. `https://shiny.crunch.io/user/{userid}/{app_dir}/` where userid and app_dir are replaced with your information. Then, in the webapp, open the dataset you want to display the cruncy app in. Once the dataset is open, click on the name of the dataset, then click Configure Dashboard. A menu will slide out from the left. Click on the button next to `URL`, then in the adjacent text box, paste in the actual url that you copied in the first step, then click Save. The page should refresh with your crunchy app.

Next: Crunch internals